

Brooklyn College
Department of Computer and Information Science

CISC 3130 [22] Data Structures

4 hours; 4 credits

Stacks and their implementations. Prefix, postfix, and infix notation. Queues and linked lists and their implementations. Binary and general trees and their implementations and traversals. Sorting and searching techniques. Graph algorithms.

Objectives

By the end of the course, students should be able to:

1. Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, and trees and be able to use these structures effectively in application programs.
2. Implement various data structures in more than one manner, compare the different implementations and explain the advantages and disadvantages of the different implementations.
3. Demonstrate understanding of and be able to program various sorting algorithms, and be able to compare the efficiency of these algorithms in terms of both time and space.
4. Trace and code recursive functions.
5. Demonstrate some understanding of object-oriented programming and be able to program with C++ classes.

Textbook

Data Structures Using C and C++ second edition, by: Langsam, Tenenbaum, and Augenstein

Syllabus

- 1 Introduction, review parameter transmission, introduction to prefix and postfix notation
- 2 review notation, start to implement stacks
- 3 implementing a stack using a structure
- 4 overflow, underflow, introduction to recursion
- 5 review stack implementation, discuss queues, recursion via stack for locals, params

- 6 introduction to C++, separate compilation
- 7 stacks in C++
- 8 discussion of nodes, up to struct node
- 9 C implementation of a stack using nodes
- 10 C++ stack implementation via Node and Stack classes
- 11 queue primitive ops, implementation in C++
- 12 implementation using array of nodes, review for test
- 13 TEST 1
- 14 implementing a list in C++
- 15 review nodes, lists, reference parameters in C++
- 16 list iterators, special list features
- 17 sorted list, intro to trees
- 18 implementing a tree in C++, traversal methods
- 19 non-recursive tree traversal methods
- 20 C++ implementation of a tree, general trees
- 21 review trees, intro to sorting
- 22 heapsort, radix sort, review others
- 23 big-O notation, divide and conquer
- 24 review big-O, sorts, heapsort & quicksort code
- 25 TEST 2
- 26 searching
- 27 hashing
- 28 spanning trees
- 29 review for final